AFRL-RI-RS-TR-2017-173

THE BERKELEY DATA ANALYSIS SYSTEM (BDAS): AN OPEN SOURCE PLATFORM FOR BIG DATA ANALYTICS

UNIVERSITY OF CALIFORNIA, BERKELEY

SEPTEMBER 2017

FINAL TECHNICAL REPORT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE

■ **AIR FORCE MATERIEL COMMAND** ■ **UNITED STATES AIR FORCE** ■ **ROME, NY 13441**

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09. This report is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RI-RS-TR-2017-173 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE CHIEF ENGINEER:

/ S /
MICHAEL J. MANNO
Work Unit Manager

/ S /
MICHAEL J. WESSING
Deputy Chief, Information Intelligence
Systems and Analysis Division
Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
SEP 2017	FINAL TECHNICAL REPORT	SEP 2012 – MAR 2017

4. TITLE AND SUBTITLE

THE BERKELEY DATA ANALYSIS SYSTEM (BDAS): AN OPEN SOURCE PLATFORM FOR BIG DATA ANALYTICS

5a. CONTRACT NUMBER
FA8750-12-2-0331

5b. GRANT NUMBER
N/A

5c. PROGRAM ELEMENT NUMBER
62702E

6. AUTHOR(S)
Ion Stoica, Michael Franklin, Michael Jordan, Armando Fox, Anthony Joseph, Michael Mahoney, Randy Katz, David Patterson, Scott Shenker

5d. PROJECT NUMBER
XDAT

5e. TASK NUMBER
A0

5f. WORK UNIT NUMBER
15

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of California, Berkeley
2150 Shattuck Ave Rm 313
Berkeley, CA 94704-5930

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory/RIED
525 Brooks Road
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/RI

11. SPONSOR/MONITOR'S REPORT NUMBER
AFRL-RI-RS-TR-2017-173

12. DISTRIBUTION AVAILABILITY STATEMENT
Approved for Public Release; Distribution Unlimited. This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The goal of this proposal was to deliver a modular open-source software stack that can support a new generation of large-scale analytic tools that provide answers over arbitrarily large datasets. This work was carried out by Berkeley's AMPLab, a research lab consisting of eleven faculty members and over 40 students. In addition to this grant, AMPLab (which ended in December 2016) was supported by industry affiliates and an NSF Expeditions grant. This grant was instrumental in improving our software stack, Berkeley Data Analytic System (BDAS), so that it can serve as a platform for the broader community. In particular, this grant enabled us to implement significant portions of the code-bases, integrate BDAS with commonly used tools, and make BDAS much easier to manage. In addition, it allowed us to extend the functionality of BDAS in several key area, including streaming, and query processing. Thanks to xData, BDAS has enjoyed a big success both in academia and industry. Today, Apache Spark is used by thousands of companies in production and counts over 400K meetup members worldwide, while Apache Mesos and Alluxio (formerly known as Tachyon) are used by hundreds of companies around the world.

15. SUBJECT TERMS
Data Management, Data Analytics, Machine Learning, Cluster Computing, Cloud Computing, Crowdsourcing, Big Data

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			**MICHAEL J. MANNO**
U	U	U	UU	29	19b. TELEPHONE NUMBER (*Include area code*)

Table of Contents

Section **Page**

Abstract
The Berkeley Data Analysis System (BDAS): An Open Source Platform for Big Data Analytics

Ion Stoica, Michael Franklin, Michael Jordan, Armando Fox, Anthony Joseph,
Michael Mahoney, Randy Katz, David Patterson, Scott Shenker

UC Berkeley

While we are awash in a sea of Big Data, this data is only as valuable as the answers it provides. Our proposed effort will deliver a modular open-source software stack that can support a new generation of large-scale analytic tools that provide answers over arbitrarily large datasets within a user's cost, time, and quality constraints. The work will be carried out by Berkeley's AMPLab, a research lab consisting of eleven faculty members and over 30 students. AMPLab is currently supported by industry affiliates and an NSF Expeditions grant. The work proposed herein is distinct from but complementary to the work supported by NSF, and is organized around two main goals.

The first goal is to make the software stack we are currently developing - the Berkeley Data Analytic System (BDAS) | more widely usable, so that it can serve as a platform for the broader community. As happened with BSD many years ago, and more recently with motes (for sensor-nets), we believe the availability of a generally applicable and well- supported open-source data analytics platform can have a transformative effect on the Big Data research community, creating a new level of synergy (through the use of common tools) and participation (by lowering the barriers to entry for research groups). To make our current BDAS stack usable by the larger community, we need to: (1) re-implement significant portions of the codebases; (2) build a sophisticated large-scale debugger; (3) integrate BDAS with commonly used tools; and (4) make BDAS much easier to manage. Our current funding does not support these kinds of engineering tasks, yet they are exactly what is needed to make BDAS a community-wide platform.

The second goal is to extend the functionality of BDAS in several key area. In particular, we will provide support for: streaming, because many data analysis tasks must process data as it arrives, rather than working from previously collected data; sampling, because as datasets grow ever larger, we can only process a subset within a given time limit; asynchronous algorithms, because they are more efficient to implement and converge faster; and a generalized form of query planning that will allow BDAS to make use of a heterogeneous infrastructure and make smart tradeoffs between cost, quality, and timeliness. All of these efforts go beyond what was envisioned in the NSF project.

1 EXECUTIVE SUMMARY

In most military situations we now have at our disposal an unprecedented volume of data coming from ever-increasing variety of sources, ranging from sensors to satellites to social networking sites. However, our ability to collect data has far outstripped our ability to process that data into useful information. This represents a fundamental challenge because our current data analytic techniques simply cannot cope with data at this scale. Turning data into information is how we make sense of the world, and it is imperative that we discover how to make sense at scale.

To meet this challenge, we propose to build the Berkeley Data Analytic System (BDAS). BDAS involves rethinking every aspect of the data analytic task with the challenges of Big Data in mind. Our preliminary experiences with components of BDAS suggest that such a redesign will increase the speed and scalability by an order of magnitude or more. BDAS will also allow flexible tradeoffs between time, quality, and cost. As such, BDAS will revolutionize the way data analytics is done.

BDAS will be open-source, with the goal of nucleating a broader community effort in this area. BDAS's modular design will mitigate risk and maximize upside by allowing third-party components to supply missing functionality and leverage external innovations. We propose a 54-month effort, with a total cost of $9 million. We expect to have a widely usable system within the first year that can serve as the platform for future innovations from other groups as well as our own effort.

2 INTRODUCTION

Extracting useful knowledge from data has long been one of the grand challenges of computer science. Indeed, entire subfields of our discipline, from machine learning to data mining to information retrieval to databases, are devoted to this task. While progress along these traditional lines of inquiry continues, the more general problem of data analysis is being transformed almost beyond recognition by four recent trends, which the popular press lumps together under the term Big Data.

• **Cloud computing:** No longer do sophisticated data analytic computations require large budgets and dedicated computational facilities. Instead, warehouse-scale computers (WSCs) provide staggering amounts of computational power that can be accessed over the Internet. Anyone with a modest budget can run large-scale data analytic computations.

• **Massive and diverse data**: The volume of data being put online is staggering. Sensors record environmental conditions and other quantities at unprecedented levels of detail, and the advent of participatory sensing promises to accelerate this trend. Simulations, scientific measurements, satellite images, and energy monitors are also providing increasingly detailed datasets. Social networks are a rich source of data, and streams of tweets, blogs, photos, and videos identify breaking events faster and in more detail than ever before. Companies are recording customer data at an increasingly fine granularity, creating both the opportunity and the need for recovering value from these large datasets.

• **Diverse and time-sensitive queries**: The increasing heterogeneity of the data has led to a far more diverse set of queries. Thus, data analytics systems cannot be optimized for a narrow set of

tasks, but instead must be far more general in nature. In addition, data analysis has become deeply entrenched in organizational decision-making, making the timeliness of queries increasingly important.

The explosion of available data means that — in areas as disparate as medicine, business, statecraft, and homeland defense — there is an unprecedented opportunity to make data-driven decisions that can significantly improve our quality of life. However, the straightforward application of previously-existing data analytics technology would not enable users to obtain timely, cost-effective answers of sufficient-quality.

Meeting this challenge requires a new data analytics paradigm that involves three dimensions:

• **Algorithms**: Improving the scale and efficiency of machine learning (ML) and data analytics algorithms, while making confidence estimation ("error bars") an integral part of all aspects of the data analytics process.

• **Machines**: Creating a more flexible datacenter infrastructure that can apply the huge scale of WSCs to a wide variety of computational problems while lowering the cost of using those resources.

• **People**: Leveraging human activity and intelligence to deal with cases that are ML-hard (i.e., cases where automated data analysis on its own does not produce a high-confidence answer).

Our research summarized in this report is part of a larger effort that we call the Algorithms, Machines and People (AMP) Project. Rather than viewing these three research thrusts as independent and abstract, the AMP project will bring them together to develop a unified architecture. The goals of this seedling project are: 1) to better understand the requirements of such a system, 2) to develop an initial proposal for such an architecture, and 3) to build and experiment with prototypes of some of the key components of the architecture.

An important part of this project is an open-source software stack known the Berkeley Data Analytics System (BDAS), which we will briefly describe below.

Before doing that, however, we should note that, while the popular coverage of Big Data has focused mostly on the size of the data, we see the requirements for a large scale analytics system such as BDAS as arising from four separate factors: the size of datasets, the rate of growth of those datasets, the heterogeneity and variable quality of data, and the resulting diversity of queries.

• *The size of datasets*: WSCs will be necessary to handle the sheer volume of data, but WSCs scale through the use of parallelism – using both multiple machines (on the order of tens of thousands) and multiple cores per machine (soon to reach dozens) – so data analysis algorithms must be rethought and rewritten to leverage extreme degrees of parallelism and powerful languages and tools must be provided to implement these algorithms.

• *The growth of datasets*: For many datasets, the rate of growth far exceeds the cost-performance improvements in computation and storage; while it is easy to tap additional data sources, it is hard

to know what data can be safely thrown away, so dataset growth is typically accelerating (i.e., the rate at which datasets grow is increasing). The important implication of this dataset acceleration is that no matter what the absolute size of an initial dataset, users are forced to devote an ever-increasing share of their resources to storing and analyzing their data.

• *Heterogeneity and variable quality of data*: A challenge that may be even more fundamental than size and growth is the heterogeneous, incomplete, and often conflicting nature of data. Such "dirty" data will soon become the norm, not the exception, because data is now coming from a wide variety of sources, leading to significant duplication and overlap, as well as variations in schemas, quality, and provenance.

• *Diversity of queries and users*: The factors outlined above mean that datasets will support a wide variety of queries, and this poses two challenges. First, it is hard to provide reasonable error bounds over a broad class of queries, where testing many hypotheses is likely to turn up a false positive because by chance one hypothesis happens to show success on that particular dataset. Second, with data analytics being applied to many new problems across a range of scales and domains, a variety of computational methods must be used.

WSC programming models prior to our project target relatively small classes of analytics tasks and users, and we must define more general datacenter programming abstractions in order to cope with the increasing diversity of tasks. For example, Pregel analyzes graphs, Map Reduce/Hadoop has rather limited flexibility, and more recent programming frameworks – such as Dryad and HaLoop – while more flexible than previously widely-used frameworks, still fall far short of what is needed.

Thus, while the size of Big Data is a major concern, there are other important considerations: the growth of data makes the Big Data problem universal, the dirtiness of data challenges traditional notions of data integration, and the diversity of queries and users raises the need for a more flexible and powerful computational framework. All of these issues must be considered in the design and development of the BDAS system.

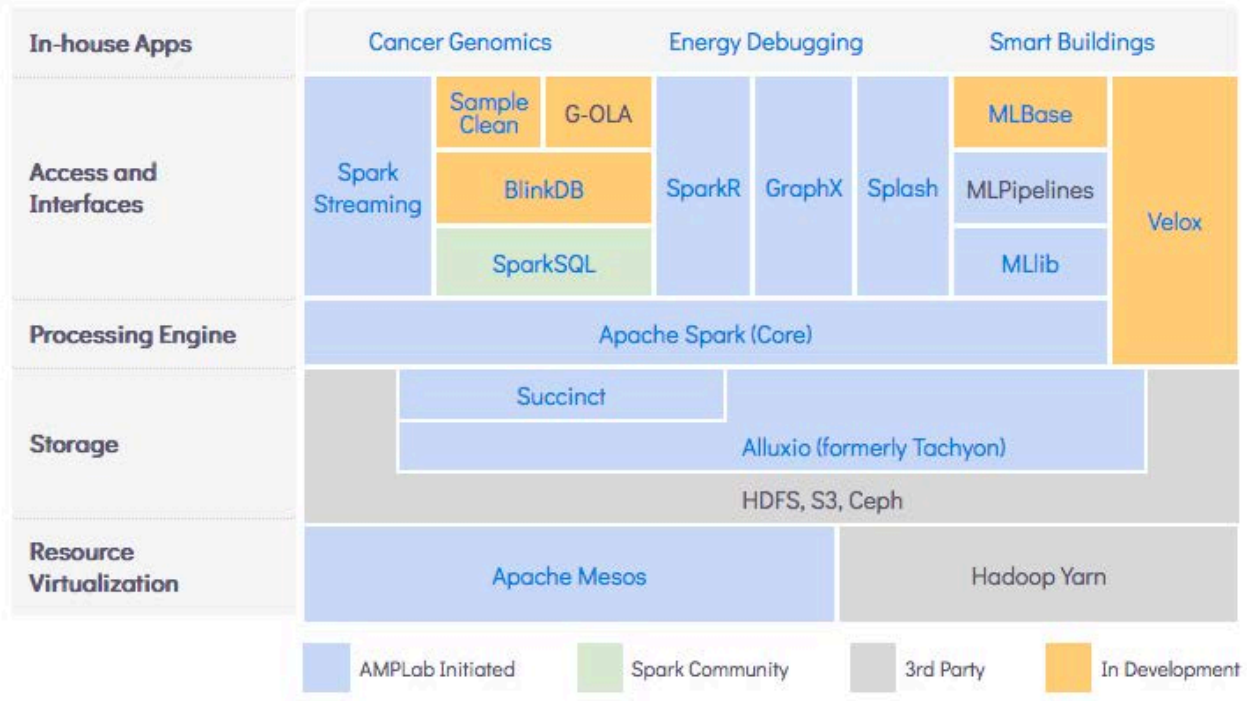
3 METHODS, ASSUMPTIONS, AND PROCEDURES

This project included both a basic research component as well as a full development and deployment component, including open source implementations of the basic research component. State-of-the-art methods were used that included: requirements analysis, component specification, rapid prototyping and measurement on cloud-based infrastructure such as Amazon Web Services and local clusters as well as crowdsourcing platforms such as Amazon Mechanical Turk, and a home-grown system we developed called CrowdDB. Details are provided in the next section as appropriate under different results.

4 RESULTS AND DISCUSSIONS

4.1 Software: A High-Level Architecture for BDAS

The software stack developed is known as BDAS. *BDAS, the Berkeley Data Analytics Stack*, is an open source software stack that integrates software components being built by the AMPLab to make sense of Big Data. BDAS consists of the components shown below. More details on many of these components are described in the following sections.



In addition to BDAS, the AMPLab has released additional software components useful for processing data. For example:

- AMPCrowd: A RESTful web service for sending tasks to human workers on crowd platforms like Amazon's Mechanical Turk. Used by the SampleClean project for context-heavy data cleaning tasks.

4.2 Community

More generally, AMPLab has been engaged in aggressive outreach, training, and communication to develop a user community around our software products. Examples of this include the following:

- Software project Meetups – Help organize monthly developer meetups around BDAS components. Check out the Spark/Shark meetup group, the Mesos meetup group, and the Alluxio meetup group

- AMP Camp "Big Data Bootcamp" – Two days packed full of software system intros, demos and hands-on exercises. Aims to bring practitioners with no prior experience up to speed and writing real code with real advanced algorithms.
- Support – Unlike many research software prototypes that never see production use, we support BDAS software components by actively monitoring and responding on developer and user mailing lists.

4.3 Alluxio (formerly Tachyon), a Memory Speed Virtual Distributed Storage System

As datasets continue to grow, storage and networking pose the most challenging bottlenecks for many workloads. To address the bottleneck, we developed Alluxio (formerly known as Tachyon), a memory-centric, fault-tolerant virtual distributed storage system. With Alluxio, any application can access any data from anywhere. Any application can store any data to anywhere.

Alluxio is memory centric — not just memory only — and its tiered storage feature means it can access any storage media. Because Alluxio provides a storage unification layer through an API, applications can access any underlying persistent storage and file systems. Alluxio supports any big data framework (Apache Spark, Apache MapReduce, Apache Flink, Impala, etc.) with any storage system or file system (Alibaba OSS, Amazon S3, EMC, NetApp, OpenStack Swift, Red Hat GlusterFS, and more), running on any storage media (SSD, HDV, DRAM, etc.).

The Alluxio project is open source (Apache 2 license) and is already deployed in production clouds on Petabyte-scale workloads. The project is the storage layer of the Berkeley Data Analytics Stack (BDAS) and also part of the Fedora distribution. Collaborating with our industry partners, the lab is continuously enhancing the system and developing exciting things around it.

For more information, please visit the Alluxio Website. The source code can be obtained from the project's Github. We also host regular meetup at the Bay Area. The project is commercially backed by Alluxio, Inc.

4.4 Cancer Tumor Genomics: Fighting the Big C with the Big D

It may have been true once that expertise in computer science was needed only by computer scientists. But Big Data has shown us that's no longer the case. The war against cancer is increasingly moving into cyberspace, and it is entirely possible that we have the skill sets needed now to fight cancer.

The cost of turning pieces of DNA into digital information has dropped more than a hundredfold in the last three years. Given such dramatic improvement, we could soon afford to sequence the genomes of the millions of cancer patients, which only billionaires could afford just a few years ago. To make personalized medicine affordable for everyone, we need to drive down the information processing costs.

AMP technology could help. The war needs new algorithms to find those needles in haystacks that could help cancer patients find effective therapies based on their genetics. To process genome data

faster and more cheaply, the war needs new infrastructure to use many machines in the cloud simultaneously. And it needs to be able to engage the wisdom of the crowd when the problems of cancer genome discovery and diagnosis are beyond our algorithms and machines.

The Cancer Tumor Genome project has already had a breakthrough that creates full genomes by aligning reads from sequencing machines 10 to 100 times faster.
This project is guided by the following observation:

"Given that millions of people do have and will get cancer, if there is a *chance* that computer scientists may have the best skill set to fight cancer today, as moral people aren't we *obligated* to try?"

To learn more, see this essay from the *New York Times*.

4.5 CoCoA: A Framework for Distributed Optimization

A major challenge in many large-scale machine learning tasks is to solve an optimization objective involving data that is distributed across multiple machines. In this setting, optimization methods that work well on single machines must be re-designed to leverage parallel computation while reducing communication costs. This requires developing new distributed optimization methods with both competitive practical performance and strong theoretical convergence guarantees. CoCoA is a novel framework for distributed computation that meets these requirements, while allowing users to re-use arbitrary single machine solvers locally on each node.

4.6 Concurrency Control for Machine Learning

Many machine learning (ML) algorithms iteratively transform some global state (e.g., model parameters or variable assignment) giving the illusion of serial dependencies between each operation. However, due to sparsity, exchangeability, and other symmetries, it is often the case that many, but not all, of the state-transforming operations can be computed concurrently while still preserving serializability: the equivalence to some serial execution where individual operations have been reordered.

This opportunity for serializable concurrency forms the foundation of distributed database systems. In this project, we implement updates in ML algorithms as concurrent transactions in a distributed database. As a result, we achieve high scalability while maintaining the semantics and theoretical properties of original serial algorithm.

ML algorithms that have been implemented using concurrency control include non-parametric clustering, correlation clustering, submodular maximization, and sparse convex optimization.

4.7 CrowdDB – Answering Queries with Crowdsourcing

CrowdDB uses human input via crowdsourcing to process queries that neither database systems nor search engines can adequately answer. It uses SQL both as a language for posing complex queries and as a way to model data. While CrowdDB leverages many aspects of traditional

database systems, there are also important differences. Conceptually, a major change is that the traditional closed-world assumption for query processing does not hold for human input. From an implementation perspective, human-oriented query operators are needed to solicit, integrate and cleanse crowdsourced data. Furthermore, performance and cost depend on a number of new factors including worker affinity, training, fatigue, motivation and location.

CrowdDB is being researched and built in collaboration with researchers at ETH Zurich. An overview paper is available here. CrowdDB wins the Best Demo award at VLDB 2011.

4.8 DNA Processing Pipeline

Another effort related to genomics underway at the AMP Lab involves developing a variant calling pipeline. Variant calling is the process of translating the output of DNA sequencing machines, short reads, to a summary of the unique characteristics of the individual being sequenced, variants. Variants are reported as differences between the individual and a reference genome.

SNAP, another AMP Lab project, is the first step of this pipeline. SNAP performs sequence alignment, whereby each short read is assigned a location of the reference genome which it closely matches. The rest of the variant calling pipeline combines the information scattered through the aligned reads into a complete picture of the individual's unique genome.

A second is a new format for storing genomic in called ADAM. ADAM is a cluster friendly storage format for genetic information that embraces modern systems technology to accelerate other steps of the genomic processing software pipeline. For example, ADAM executes two of the most expensive steps 110 times faster using an 82-node cluster.

Another expensive step in a genomics pipeline is identifying the differences between the standard human reference and each person, named *variant calling*. Alas, it is slow, taking hundreds of hours per genome. Papers proposing new variant callers typically use unique data sets and metrics, as genetics benchmarks do not exist. Hence, an important question for pipelines is performance; i.e., how accurate are the variants that they call? This is difficult to determine when it is applied to real data, since we don't know the ground truth. Thus, we are developing a suite of benchmarks for evaluating variant calling pipelines called. SMaSH, which is a variant calling benchmark suite with appropriate evaluation metrics. As there is no real ground truth for genetics—the technology cannot yet specify 3B base pairs perfectly—it is trickier than in CS. Just as CS fields accelerate when benchmarks are embraced, we hope that SMaSH will accelerate variant calling.

This pipeline is an important part of our effort regarding cancer genomics. To analyze a tumor and identify important mutations that are relevant to choosing a treatment, the raw output of a DNA sequencing machine must be processed via this pipeline.

4.9 DNA Sequence Alignment with SNAP

As the cost of DNA sequencing continues to drop faster than Moore's Law, there is a growing need for tools that can efficiently analyze larger bodies of sequence data. By mid-2013, sequencing a human genome is expected to cost $1000, at which point this technology enters the realm of

routine clinical practice. For example, it is expected that each cancer patient will have their genome and their cancer's genome sequenced. Assembling and interpreting the short read data produced by sequencers in a timely fashion, however, is a significant challenge, with current pipelines taking thousands of CPU-hours per genome.

Here, we address the first and most expensive step of this process: aligning reads to a reference genome. We present the Scalable Nucleotide Alignment Program (SNAP), a new aligner that is 10-100x faster and simultaneously more accurate than existing tools like BWA, Bowtie2 and SOAP2. Unlike recent aligners that use graphical processing units (GPUs), SNAP runs on commodity processors. Furthermore, whereas existing fast aligners limit the number and types of differences from the reference genome they allow per read, SNAP supports a rich error model and can cheaply match reads with more differences. This gives it up to 2x lower error rates than current tools and lets it match classes of mutations, such as longer indels, that these tools miss.

SNAP is open source on the SNAP homepage. A technical report on the algorithm is also available. SNAP is a joint project with Microsoft Research and UC San Francisco.

4.10 GraphX: Large-Scale Graph Analytics

Increasingly, data-science applications require the creation, manipulation, and analysis of large graphs ranging from social networks to language models. While existing graph systems (e.g., GraphBuilder, Titan, and Giraph) address specific stages of a typical graph-analytics pipeline (e.g., graph construction, querying, or computation), they do not address the entire pipeline, forcing the user to deal with multiple systems, complex and brittle file interfaces, and inefficient data-movement and duplication.

The GraphX project unifies graphs and tables enabling users to express an entire graph analytics pipeline within a single system. The GraphX interactive API makes it easy to build, query, and compute on large distributed graphs. In addition, GraphX includes a growing repository of graph algorithms for a range of analytics tasks. By casting recent advances in graph processing systems as distributed join optimizations, GraphX is able to achieve performance comparable to specialized graph processing systems while exposing a more flexible API. By building on top of recent advances in data-parallel systems, GraphX is able to achieve fault-tolerance while retaining in-memory performance and without the need for explicit checkpoint recovery.

GraphX is available as part of the Spark Apache Incubator project as of version 0.9.0, and the active research version of GraphX can be obtained from the GitHub project page.

4.11 KeystoneML

KeystoneML is a research project exploring techniques to simplify the construction of *large scale*, *end-to-end*, and machine learning pipelines.

KeystoneML is designed around the principles of composability and modularity, and presents a rich set of operators including featurizers for images, text, and speech, as well as general purpose statistical and signal processing tools including large scale linear solvers. The software also

provides several example pipelines that reproduce state-of-the-art academic results on public data sets.

KeystoneML is open source software built on Apache Spark. You can find more information and examples on the project webpage, and contribute to the code on Github.

4.12 MDCC: Multi-Data Center Consistency

MDCC (Multi-Data Center Consistency) is a project to efficiently achieve stronger consistency for databases deployed across several different data centers. MDCC has two main components: A new Service-level-objective aware programming model which empowers the developer with more information about the transaction, and a new latency-aware commit protocol which commits most transactions with a single round trip message delay. Visit the MDCC webpage for more details!

4.13 Mesos – Dynamic Resource Sharing for Clusters

Mesos is a cluster manager that provides efficient resource isolation and sharing across distributed applications, or *frameworks*. It can run Hadoop, MPI, Hypertable, Spark (a new framework for low-latency interactive and iterative jobs), and other applications. Mesos is open source in the Apache Incubator.

More information and downloads can be found on the Mesos homepage.

4.14 MLbase: Distributed Machine Learning Made Easy

Implementing and consuming Machine Learning techniques at scale are difficult tasks for ML Developers and End Users. MLbase is a platform addressing the issues of both groups, and consists of three components: MLlib, MLI, and ML Optimizer.

For more details, please visit http://mlbase.org.

4.15 PIQL – Scale Independent Query Processing

PIQL is a SQL like language that uses a new *scale independent* optimization strategy to execute relational queries while maintaining the performance predictability and scalability provided by distributed key/value stores. Scale independent optimization guarantees that all queries will perform a bounded number of storage operations independent of the size of the underlying database. PIQL employs language extensions, query compilation technology, automatic materialized view selection and response-time estimation to provide scale independence.

More details can be found in our VLDB2012 paper.

4.16 Real Life Datacenter Workloads

How do we ensure that AMP Lab works on important and immediate problems? One of many ways is to look at real life workloads from our industry partners and their customers.

AMP Lab is fortunate to have under our analysis the activity logs of real life, front line systems of up to 1000s of nodes servicing 100s of PB of data. As of early 2012, these logs include Hadoop, Dryad, enterprise network storage, and other similar systems, from Cloudera, Facebook, Google, Microsoft, Netapp, and Twitter.

We view it as a part of our academic contribution to

- Scientifically understand these workloads,
- Improve large scale systems according to empirical behavior,
- Share our insights with the research community,
- Help our industry partners innovate on design and performance, and ultimately
- Train ourselves to be knowledgeable on using big data to improve the society at large.

Selected publications heavily influenced by real-life workloads (in reverse publication order):

- Interactive Analytical Processing in Big Data Systems: A Cross Industry Study of MapReduce Workloads (in press)
- Understanding TCP Incast and Its Implications for Big Data Workloads
- PACMan: Coordinated Memory Caching for Parallel Jobs
- Energy Efficiency for Large-Scale MapReduce Workloads with Significant Interactive Analysis
- Design Implications for Enterprise Storage Systems via Multi-Dimensional Trace Analysis
- The Case for Evaluating MapReduce Performance Using Workload Suites
- Disk-Locality in Datacenter Computing Considered Irrelevant
- Design and Evaluation of a Real-Time URL Spam Filtering Service
- Scarlett: Coping with Skewed Popularity Content in MapReduce Clusters
- Mesos: A Platform for Fine-Grained Resource Sharing in the Data Center
- Dominant Resource Fairness: Fair Allocation of Multiple Resources Types
- Reining in the Outliers in MapReduce Clusters using Mantri
- Privacy Settings in Context: A Case Study using Google Buzz

4.17 SampleClean: Fast and Accurate Query Processing on Dirty Data

In emerging Big Data scenarios, obtaining timely, high-quality answers to aggregate queries is difficult due to the challenges of processing and cleaning large, dirty data sets. To increase the speed of query processing, there has been a resurgence of interest in sampling-based approximate query processing (SAQP). In its usual formulation, however, SAQP does not address data cleaning at all, and in fact, exacerbates answer quality problems by introducing by sampling error. We explore the use of sampling to actually improve answer quality. We introduce the Sample-and-Clean framework, which applies data cleaning to a relatively small subset of the data and uses the results of the cleaning process to lessen the impact of dirty data on aggregate query answers.

More details and downloads can be found on the SampleClean homepage.

4.18 Spark – Lightning-Fast Cluster Computing

Spark is an open source cluster computing system that aims to make data analytics *fast* — both fast to run and fast to write. To run programs faster, Spark provides primitives for in-memory cluster computing: your job can load data into memory and query it repeatedly much quicker than with disk-based systems like Hadoop MapReduce. To make programming faster, Spark integrates into the Scala programming language, letting you manipulate distributed datasets like local collections. You can also use Spark interactively to query big data from the Scala interpreter.

More details and downloads can be found on the Spark homepage.

4.19 Sparrow: Low Latency Scheduling for Interactive Cluster Services

The Sparrow project introduces a distributed cluster scheduling architecture which supports ultra-high throughput, low latency task scheduling. By supporting very low-latency tasks (and their associated high rate of task turnover), Sparrow enables a new class of cluster applications which analyze data at unprecedented volume and speed. The Sparrow project is under active development and maintained in our public GitHub repository.

4.20 Splash: Efficient Stochastic Learning on Clusters

Splash is a general framework for parallelizing stochastic learning algorithms (SGD, Gibbs sampling, etc.) on multi-node clusters. It consists of a programming interface and an execution engine. You can develop any sequential stochastic algorithm using the programming interface without considering the underlying distributed computing environment. The only requirement is that the base algorithm must be capable of processing *weighted samples*. The parallelization is taken care of by the execution engine and is communication efficient. Splash is built on Apache Spark and is closely integrated with the Spark ecosystem.

On large-scale datasets, Splash can be substantially faster than existing data analytics packages built on Apache Spark. For example, to fit a 10-class logistic regression model on the mnist8m dataset, stochastic gradient descent (SGD) implemented with Splash is 25x faster than MLlib's L-BFGS and 75x faster than MLlib's mini-batch SGD for achieving the same accuracy. All algorithms run on a 64-core cluster. To learn more about Splash, visit the Splash website or read our paper.

4.21 Succinct: Enabling Queries on Compressed Data

Web applications and services today collect, store and analyze an immense amount of data. As data sizes continue to grow, the bottlenecks in systems for big data analytics have undergone a fundamental change. In particular, with memory bandwidth and CPU performance growing at a rate much faster than the bandwidth between CPU and slower storage devices (SSD, disk, etc.), existing big data systems are increasingly bottlenecked by I/O. These I/O bottlenecks are (and will continue to be) getting worse!

A fundamental approach to alleviating the I/O bottlenecks is to use data compression. Traditional compression techniques have led to significant gains in terms of storage costs, energy costs, and performance for a wide variety of batch processing jobs. These techniques have also been used for reducing I/O bottlenecks in columnar stores with significant performance improvements for OLAP workloads that typically require scanning the entire dataset.

However, the aforementioned compression and query execution techniques are unsuitable for a wide variety of workloads that do not necessarily require data scans (e.g., point queries). One example is search, a fundamental primitive supported by many web applications and services. Examples include Facebook search, Twitter search, LinkedIn search, airline and hotel search, and services that are specifically built around search (Google, Bing, Yelp, to name a few). Another example is random access as typically performed via get interface in key-value stores, NoSQL stores, document stores, etc. Queries in such workloads are often short-lived (ideally sub-millisecond), and data scans and/or decompression are not useful for such short-lived queries. Given the large number of applications that run such workloads, we at AMPLab decided to take a stab at this problem and asked the following fundamental question:

Is it possible to execute point queries (e.g., search and random access) directly on compressed data without performing data scans?

Exploring the above question led to the Succinct project! At a high-level, Succinct enables a wide range of queries including search, range and wildcard queries over arbitrary strings as well as random access into the input data directly on a compressed representation of the input. What differentiates Succinct from previous systems that support point queries is that Succinct supports these queries without storing any indexes, without data scans and without data decompression — all the required information is embedded within the compressed representation and queries are executed directly on the compressed representation.

On real-world and benchmark datasets, Succinct can execute sub-millisecond search queries while keeping as much as an order of magnitude more input data in faster storage compared to state-of-the-art systems that provide similar functionality using indexes. For example, on a server with 128GB RAM, Succinct can push as much as 163 — 250GB of raw data, depending on the dataset, while executing search queries within a millisecond. Thus, Succinct executes more queries in faster storage, leading to lower query latency than existing systems for a much larger range of input sizes.

For more information on Succinct — techniques, tradeoffs and benchmark results— see the Succinct webpage. A good place to start experimenting with Succinct is Succinct on Apache Spark, an Apache Spark package that enables queries directly on compressed RDDs. There are a large number of interesting follow up projects in AMPLab on Succinct exploring the fundamental limits to querying on compressed data, adding new applications on top of Succinct, and improving the performance for existing applications. We will write a lot more about these very exciting projects on Succinct webpage.

4.22 Velox: Models in Action

To support complex data-intensive applications such as personalized recommendations, targeted advertising, and intelligent services, the data management community has focused heavily on the design of systems to support training complex models on large datasets. Unfortunately, the design of these systems largely ignores a critical component of the overall analytics process: the deployment and serving of models at scale. We present Velox, a new component of the Berkeley Data Analytics Stack. Velox is a data management system for facilitating the next steps in real-world, large-scale analytics pipelines: online model management, maintenance, and prediction serving. Velox provides end-user applications and services with a low-latency, intuitive interface to models, transforming the raw statistical models currently trained using existing offline large-scale compute frameworks into full-blown, end-to-end data products. To provide up-to-date results for these complex models, Velox also facilitates lightweight online model maintenance and selection (i.e., dynamic weighting). Velox has the ability to span online and offline systems, to adaptively adjust model materialization strategies, and to exploit inherent statistical properties such as model error tolerance, all while operating at "Big Data" scale.

Check out the code on Github or our paper in CIDR 2015.

5 CONCLUSIONS

At the AMPLab, we have developed methods to help turn data into knowledge, thereby making sense of the world around us. Computer science is now on the verge of a new era in data analysis because of several recent developments, including: the rise of the warehouse-scale computer (WSC), the massive explosion in online data; the increasing diversity and time-sensitivity of queries; the advent of crowdsourcing; and so on. Together these and other trends — often referred to collectively as Big Data — have the potential for ushering in a new era in data analysis; but to realize this opportunity requires us to confront several significant scientific challenges. These challenges include that: the programming environments developed for these WSCs are only effective on a narrow range of tasks; massive data typically come from diverse sources with no common schema and are of variable quality; we need far more flexible, scalable, and tunable analysis algorithms so that, over a wide range of queries, explicit tradeoffs can be made between delay, cost, and quality-of-answer. Meeting these and other challenges require an entirely new approach that transcends and reshapes disciplinary boundaries. The AMPLab was a five-year collaborative effort at UC Berkeley, involving students, researchers and faculty from a wide swath of computer science and data-intensive application domains to address the Big Data analytics problem. We have made major advances that have been widely adopted.

6 REFERENCES

[1] Andrew Wang, Shivaram Venkataraman, Sara Alspaugh, Randy Katz, and Ion Stoica. Cake: Enabling High-level SLOs on Shared Storage Systems. ACM Symposium on Cloud Computing (SOCC), 2012. https://sites.google.com/site/acm2012socc/papers.

[2] Charles Reiss, Alexey Tumanov, Gregory R. Ganger, Randy H. Katz, and Michael A. Kozuch. Heterogeneity and Dynamicity of Clouds at Scale: Google Trace Analysis. ACM Symposium on Cloud Computing (SOCC), 2012.
https://sites.google.com/site/acm2012socc/papers.

[3] Peter Bailis, Alan Fekete, Ali Ghodsi, Joseph M. Hellerstein, and Ion Stoica. The Potential Dangers of Causal Consistency and an Explicit Solution. ACM Symposium on Cloud Computing (SOCC), 2012. https://sites.google.com/site/acm2012socc/papers.

[4] Ganesh Anantharanayanan, Christopher Douglas, Raghu Ramakrishnan, Sriram Rao, and Ion Stoica. True Elasticity in Multi-Tenant Clusters through Amoeba. ACM Symposium on Cloud Computing (SOCC), 2012.

[5] Adam Oliner, Anand Padmanabha Iyer, Eemil Lagerspetz, Sasu Tarkoma, Ion Stoica. Carat: Collaborative Energy Debugging for Mobile Devices. USENIX HotDep '12, 2012.

[6] Mosharaf Chowdhury and Ion Stoica. Coflow: An Application Layer Abstraction for Cluster Networking. ACM Workshop on Hot Topics in Networks (Hotnets IX), 2012.

[7] Sameer Agarwal (UC Berkeley), Barzan Mozafari (MIT), Aurojit Panda (UC Berkeley), Henry Milner (UC Berkeley), Sam Madden (MIT), Ion Stoica (UC Berkeley). MLbase: A Distributed Machine-learning System. CIDR, 2013.
https://amplab.cs.berkeley.edu/publication/mlbase-a-distributed-machine-learning-system/

[8] Gianluca Demartini, Beth Trushkowsky, Tim Kraska, Michael Franklin. CrowdQ: Crowdsourced Query Understanding. CIDR, 2013.
https://amplab.cs.berkeley.edu/publication/crowdq-crowdsourced-query-understanding/

[9] Tim Kraska, Gene Pang, Michael Franklin, Sam Madden, Alan Fekete. MDCC: Multi-Data Center Consistency. EuroSys '13, 2013. https://amplab.cs.berkeley.edu/wp-content/uploads/2013/03/mdcc-eurosys13.pdf

[10] Sameer Agarwal, Barzan Mozafari, Aurojit Panda, Henry Milner, Sam Madden, Ion Stoica. BlinkDB: Queries with Bounded Errors and Bounded Response Times on Very Large Data. EuroSys '13, 2013. http://www.cs.berkeley.edu/~sameerag/blinkdb_nsdi12.pdf

[11] Peter Bailis, Ali Ghodsi. Eventual Consistency Today: Limitations, Extensions, and Beyond. Communications of the ACM, 2013.
https://amplab.cs.berkeley.edu/wp-content/uploads/2013/04/p20-bailis.pdf

[12] Peter Bailis, Alan Fekete, Ali Ghodsi, Joseph M. Hellerstein, Ion Stoica. HAT, not CAP: Towards Highly Available Transactions. USENIX HotOS, 2013. https://amplab.cs.berkeley.edu/wp-content/uploads/2013/04/hat-hotos2013.pdf

[13] Kay Ousterhout, Aurojit Panda, Jashua Rosen, Shivaram Venkataraman, Reynold Xin, Sylvia Ratnasamy, Scott Shenke, Ion Stoica. The Case for Tiny Tasks in Compute Clusters. USENIX HotOS, 2013.
https://amplab.cs.berkeley.edu/wp-content/uploads/2013/04/hotos13-final24.pdf

[14] Peter Bailis, Shivaram Venkataraman, Michael Franklin, Joseph Hellerstein, Ion Stoica. PBS at Work: Advancing Data Management with Consistency Meterics. SIGMOD '13, 2013. https://amplab.cs.berkeley.edu/wp-content/uploads/2013/04/pbs-demo-sigmod2013.pdf

[15] Peter Bailis, Ali Ghodsi, Joseph Hellerstein, Ion Stoica. Bolt-on Causal Consistency. SIGMOD '13, 2013. https://amplab.cs.berkeley.edu/publication/bolt-on-causal-consistency/

[16] Reynold Xin, Joseph Gonzalez, Michael Franklin, Ion Stoica. GraphX: A Resilient Distributed Graph System on Spark. Proceedings of the First International Workshop on Graph Data Management Experience and Systems (GRADES 2013), 2013. https://amplab.cs.berkeley.edu/wp-content/uploads/2013/05/grades-graphx_with_fonts.pdf

[17] Reynold Xin, Joshua Rosen, Matei Zaharia, Michael Franklin, Scott Shenker, Ion Stoica. Shark: SQL and Rich Analytics at Scale. SIGMOD '13, 2013. https://amplab.cs.berkeley.edu/wp-content/uploads/2013/02/shark_sigmod2013.pdf

[18] Ariel Kleiner, Ameet Talwalkar, Sameer Agarwal, Ion Stoica, Michael Jordan. A General Bootstrap Performance Diagnosis. ACM KDD, 2013. https://amplab.cs.berkeley.edu/publication/a-general-bootstrap-performance-diagnostic/

[19] Sara Alspaugh, Archana Ganapathi, Marti Hearst, Randy Katz. Building Blocks for Exploratory Data Analysis Tools. ACM KDD, 2013. https://amplab.cs.berkeley.edu/publication/building-blocks-for-exploratory-data-analysis-tools/

[20] Mosharaf Chowdhury, Srikanth Kandula, Ion Stoica. Leveraging Endpoint Flexibility in Data-Intensive Clusters. ACM SIGCOMM, 2013.

[21] Kay Ousterhout, Patrick Wendell, Matei Zaharia, Ion Stoica. Sparrow: Distibuted, Low Latency Scheduling. SOSP, 2013.
https://amplab.cs.berkeley.edu/publication/sparrow-distributed-low-latency-scheduling/

[22] Matei Zaharia, Tathagata Das, Haoyuan Li, Timothy Hunter, Scott Shenker, Ion Stoica. Discretized Streams: Fault-Tolerant Streaming Computation at Scale. SOSP, 2013. http://dl.acm.org/citation.cfm?doid=2517349.2522737

[23] Evan Sparks, Ameet Talwalkar, Virginia Smith, Jey Kottalam, Xinghao Pan, Joseph Gonzalez, Michael Franklin, Michael Jordan, Tim Kraska. MLI: An API for Distributed Machine Learning. ICDM, 2013.
https://amplab.cs.berkeley.edu/publication/mli-an-api-for-distributed-machine-learning/

[24] Reyonld Xin, Dan Crankshaw, Ankur Dave, Joseph Gonzalez, Michael Franklin, Ion Stoica. GraphX: Unifying Data-Parallel and Graph-Parallel Analytics. Strata, 2014. https://amplab.cs.berkeley.edu/wp-content/uploads/2014/02/graphx.pdf

[25] Kumaripaba Athukorala, Eemil Lagerspetz, Maria von Kugelgen, Antti Jylha, Adam Oliner, Sasu Tarkoma, Giulio Jacucci. How Carat Affects User Behavior: Implications for Mobile Battery Awareness Applications. CHI (International Conference on Computer-Human Interaction), 2014. https://amplab.cs.berkeley.edu/publication/how-carat-affects-user-behavior/

[26] Liwen Sun, Michael Franklin, Sanjay Krishnan, Reynold Xin. Fine-grained Partitioning for Aggressive Data Skipping. ACM SIGMOD, 2014. https://amplab.cs.berkeley.edu/publication/fine-grained-partitioning-for-aggressive-data-skipping/

[27] Peter Bailis, Alan Fekete, Ali Ghodsi, Joseph M. Hellerstein, Ion Stoica. Scalable Atomic Visibility with RAMP Transactions. ACM SIGMOD, 2014. https://amplab.cs.berkeley.edu/publication/scalable-atomic-visibility-with-ramp-transactions/

[28] Hien Thi Thu Truong, Eemil Lagerspetz, Petteri Nurmi, Adam Oliner, Sasu Tarkoma, N. Asokan, Sourav Bhattacharya. The Company You Keep: Mobile Malware Infection Rates and Inexpensive Risk Indicators. WWW (International Conference on World Wide Web), 2014.

[29] Sameer Agarwal, Henry Milner, Ariel Kleiner, Ameet Talwalkar, Michael Jordan, Sam Madden, Barzan Mozafari, Ion Stoica. Knowing When You're Wrong: Building Fast and Reliable Approximate Query Processing Systems. ACM SIGMOD, 2014.

[30] Jiannan Wang, Sanjay Krishnan, Michael Franklin, Ken Goldberg, Tim Kraska, Tova Milo. A Sample-and-Clean Framework for Fast and Accurate Query Processing on Dirty Data. ACM SIGMOD, 2014.

[31] Gene Pang, Tim Kraska, Michael Franklin, Alan Fekete. PLANET: Making Progress with Commit Processing in Unpredictable Environments. ACM SIGMOD, 2014.

[32] Sara Alspaugh, Archana Ganapathi, Marti A. Hearst, Randy Katz. Better Logging to Improve Interactive Data Analysis Tools. ACM SIGMOD, 2014. https://amplab.cs.berkeley.edu/publication/scalable-atomic-visibility-with-ramp-transactions/

[33] Rishabh Iyer, Stefanie Jegelka, Jeff Bilmes. Monotone Closure of Relaxed Constraints in Submodular Optimization: Connections between Minimization and Maximization. Uncertainty in Artificial Intelligence, Jul. 2014.

[34] Peter Bailis, Kyle Kingsbury. The Network is Reliable: An informal survey of real-world communications failures. ACM Queue and Communications of the ACM, 2014. https://amplab.cs.berkeley.edu/publication/the-network-is-reliable/

[35] Mosharaf Chowdhury, Yuan Zhong, Ion Stoica. Efficient Coflow Scheduling with Varys. ACM SIGCOMM, 2014.
https://amplab.cs.berkeley.edu/publication/efficient-coflow-scheduling-with-varys/

[36] Liwen Sun, Sanjay Krishnan, Reynold Xin, Michael Franklin. A Partitioning Framework for Aggressive Data Skipping. Workshop on Interactive Data Exploration and Analytics (IDEA). Co-located with KDD and VLDB (Very Large Data Bases), 2014.

[37] Barzan Mozafari, Purna Sarkar, Michael Franklin, Michael Jordan, Sam Madden. Scaling Up Crowd-Sourcing to Very Large Datasets: A Case for Active Learning. VLDB (Very Large Data Bases), 2014.

[38] Peter Bailis, Aaron Davidson, Alan Fekete, Ali Ghodsi, Joseph M. Hellerstein, Ion Stoica. Highly Available Transactions: Virtues and Limitations. VLDB (Very Large Data Bases), 2014. https://amplab.cs.berkeley.edu/publication/highly-available-transactions-virtues-and-limitations/

[39] Joseph Gonzalez, Reynold Xin, Ankur Dave, Dan Crankshaw, Michael Franklin, Ion Stoica. GraphX: Graph Processing in a Distributed Dataflow Framework. USENIX Operating Systems Design and Implementation (OSDI), 2014.

[40] Sanjay Krishnan, Jay Patel, Michael Franklin, Ken Goldberg. A Methodology for Learning, Analyzing, and Mitigating Social Influence Bias in Recommender Systems. ACM Conference on Recommender Systems, 2014.

[41] Shivaram Venkataraman, Aurojit Panda, Ganesh Anantharayanan, Michael Franklin, Ion Stoica. The Power of Choice in Data-Aware Cluster Scheduling. USENIX Operating Systems Design and Implementation (OSDI), 2014.

[42] Peter Bailis, Alan Fekete, Michael Franklin, Ali Ghodsi, Joseph M. Hellerstein, Ion Stoica. Coordination Avoidance in Database Systems. VLDB 2015 (PVLDB Vol. 8 No. 3), 2014.

[43] Haoyuan Li, Ali Ghodsi, Matei Zaharia, Scott Shenker, Ion Stoica. Tachyon: Reliable, Memory Speed Storage for Cluster Computing Frameworks. ACM Symposium on Cloud Computing (SOCC), 2014.

[44] Sara Alspaugh, Betty Beidi Chen, Jessica Lin, Archana Ganapathi, Marti A. Hearst, Randy Katz. Analyzing Log Analysis: An Empirical Study of User Log Mining. Large Installation System Administration Conference (LISA), 2014.

[45] Robert Nishihara, Stefanie Jegelka, Michael Jordan. On the Convergence Rate of Decomposable Submodular Function Minimization. NIPS (Neural Information Processing Systems), 2014.

[46] Martin Jaggi, Virginia Smith, Martin Takac, Jonathan Terhorst, Sanjay Krishnan, Thomas Hofmann, Michael Jordan. Communication-Efficient Distributed Dual Coordinate Ascent. NIPS (Neural Information Processing Systems), 2014.

[47] Adarsh Prasad, Stefanie Jegelka, Dhruv Batra. Submodular meets Structured: Finding Diverse Subsets in Exponentially-Large Structured Item Sets. NIPS (Neural Information Processing Systems), 2014.

[48] Xinghao Pan, Stefanie Jegelka, Joseph Gonzalez, Joseph K. Bradley, Michael Jordan. Parallel Double Greedy Submodular Maximization. NIPS (Neural Information Processing Systems), 2014.

[49] Dan Crankshaw, Peter Bailis, Joseph Gonzalez, Haoyuan Li, Zhao Zhang, Michael Franklin, Ali Ghodsi, Michael Jordan. The Missing Piece in Complex Analytics: Low Latency, Scalable Model Management and Serving with Velox. Conference on Innovative Data Systems Research (CIDR), 2015. http://arxiv.org/pdf/1409.3809.pdf

[50] A. Vij and Kalyanaraman Shankari. When is Big Data Big Enough? Implications of Using GPS-Based Surveys for Travel Demand Analysis. Transportation Research Board 94th Annual Meeting, 2015. http://www.sciencedirect.com/science/journal/0968090X/56

[51] Yuchen Zhang, Xi Chen, Dengyong Zhou, Michael Jordan. Spectral Methods Meet EM: A Provably Optimal Algorithm for Crowdsourcing. Advances in Neural Information Processing Systems (NIPS), 2015. http://arxiv.org/abs/1406.3824

[52] Michael Armbrust, Reynold S. Xin, Cheng Lian, Yin Huai, Davies Liu, Joseph K. Bradley, Xiangrui Meng, Tomer Kaftan, Michael J. Franklin, Ali Ghodsi, Matei Zaharia. Spark SQL: Relational Data Processing in Spark. ACM SIGMOD Conference, 2015.

[53] Peter Bailis, Alan Fekete, Michael Franklin, Ali Ghodsi, Joseph M. Hellerstein, Ion Stoica. Feral Concurrency Control: An Empirical Investigation of Modern Application Integrity. ACM SIGMOD Conference, 2015.

[54] Frank Austin Nothaft, Matt Massie, Timothy Danford, Zhao Zhang, Uri Laserson, Carl Yeksigian, Jey Kottalam, Michael Franklin, Anthony Joseph, David Patterson. Rethinking Data-Intensive Science Using Scalable Analytics Systems. ACM SIGMOD Conference, 2015.

[55] Anand Padmanabha Iyer, Li Erran Li, Ion Stoica. CellIQ: Real-Time Cellular Network Analytics at Scale. 12th USENIX Symposium on Networked Systems Design and Implementation (NSDI '15), 2015. https://www.usenix.org/conference/nsdi15/technical-sessions

[56] Kay Ousterhout, Ryan Rasti, Sylvia Ratnasamy, Scott Shenker, Byung-Gon Chun. Making Sense of Performance in Data Analytics Frameworks. 12th USENIX Symposium on Networked Systems Design and Implementation (NSDI '15), 2015. https://www.usenix.org/system/files/conference/nsdi15/

[57] Kai Zeng, Sameer Agarwal, Ankur Dave, Michael Armbrust, Ion Stoica. G-OLA: Generalized Online Aggregation for Interactive Analysis on Big Data. ACM SIGMOD Conference, 2015.

[58] Neeraja Yadwadkar, Bharath Hariharan, Joseph Gonzalez, Randy Katz. Faster Jobs in Distributed Data Processing using Multi-Task Learning. SIAM International Conference on Data Mining, 2015.

[59] Xinghao Pan, Dimitris Papailiopoulos, Samet Oymak, Benjamin Recht, Kannan Ramchandran, Michael I. Jordan. Parallel Correlation Clustering on Big Graphs. arXiv, 2015.

[60] Yuchen Zhang, Martin J. Wainwright, Michael I. Jordan. Distributed Estimation of Generalized Matrix Rank: Efficient Algorithms and Lower Bounds. ICML 2015: International Conference on Machine Learning, 2015.

[61] Robert Nishihara, Laurent Lessard, Ben Recht, Andrew Packard, Michael Jordan. A General Analysis of the Convergence of ADMM. ICML 2015: International Conference on Machine Learning, 2015.

[62] Chenxin Ma, Virginia Smith, Martin Jaggi, Michael I. Jordan, Peter Richtarik, Martin Takac. Adding vs. Averaging in Distributed Primal-Dual Optimization. ICML 2015: JMLR W&CP volume37, Proceedings of the 32nd International Conference on Machine Learning, pp. 1973-1982, 2015.

[63] Benedict Paten, Matt Massie, Frank Austin Nothaft, David Patterson, David Haussler, et al. The NIH BD2K center for big data in translational genomics. Journal of the American Medical Informatics Association, 2015.

[64] Horia Mania, Xinghao Pan, Dimitris Papailiopoulos, Benjamin Recht, Kannan Ramchandran, Michael I. Jordan. Perturbed Iterate Analysis for Asynchronous Stochastic Optimization. arXiv, 2015.

[65] Daniel Haas, Jason Ansel, Lydia Gu, Adam Marcus. Argonaut: Macrotask Crowdsourcing for Complex Data Processing. 41st International Conference on Very Large Data Bases, 2015.

[66] Daniel Haas, Sanjay Krishnan, Jiannan Wang, Michael J. Franklin, Eugene Wu. Wisteria: Nurturing Scalable Data Cleaning Infrastructure. 41st International Conference on Very Large Data Bases, 2015.

[67] Sanjay Krishnan, Jiannan Wang, Michael Franklin, Ken Goldberg, Tim Kraska. Stale View Cleaning: Getting Fresh Answers from Stale Materialized Views. 41st International Conference on Very Large Data Bases, 2015.

[68] Mosharaf Chowdhury, Ion Stoica. Efficient Coflow Scheduling Without Prior Knowledge. ACM SIGCOMM, 2015.

[69] Peter Bailis, Alan Fekete, Michael Franklin, Ali Ghodsi, Joseph M. Hellerstein, Ion Stoica. Coordination Avoidance in Database Systems. 41st International Conference on Very Large Data Bases, 2015.

[70] Barzan Mozafari, Purna Sarkar, Michael Franklin, Michael Jordan, Sam Madden. Scaling Up Crowd-Sourcing to Very Large Datasets: A Case for Active Learning. 41st International Conference on Very Large Data Bases, 2015.

[71] Qifan Pu, Ganesh Anantharayanan, Peter Bodik, Srikanth Kandula, Aditya Akella, Paramvir Bahl, Ion Stoica. Low Latency, Geo-distributed Data Analytics. ACM SIGCOMM, 2015.

[72] Evan Sparks, Ameet Talwalkar, Daniel Haas, Michael Franklin, Michael Jordan, Tim Kraska. Automating Model Search for Large Scale Machine Learning. Symposium on Cloud Computing (SoCC), 2015.

[73] David Zats, Anand Padmanabha Iyer, Ganesh Anantharayanan, Rachit Agarwal, Randy Katz, Ion Stoica, Amin Vahdat. FastLane: Making Short Flows Shorter with Agile Drop Notification. Symposium on Cloud Computing (SoCC), 2015.

[74] Sanjay Krishnan, Jiannan Wang, Michael Franklin, Ken Goldberg, Tim Kraska. Stale View Cleaning: Getting Fresh Answers from Stale Materialized Views. 41st International Conference on Very Large Data Bases, 2015.

[75] Sanjay Krishnan, Animesh Garg, Sachin Patil, Colin Lea, Greg Hager, Pieter Abbeel, Ken Goldberg. Transition State Clustering: Unsupervised Surgical Trajectory Segentation for Robot Learning. ISRR 2015: The International Symposium on Robotics Research, 2015.

[76] Zhao Zhang, Kyle Barbary, Frank Austin Nothaft, Evan Sparks, Oliver Zahn, Michael J. Franklin, David A. Patterson, Saul Perlmutter. Scientific Computing Meets Big Data Technology: An Astronomy Use Case. IEEE, 2015.

[77] Daniel Haas, Jiannan Wang, Eugene Wu, Michael Franklin. Clamshell: Scaling Up Crowds for Low Latency Data Labeling. Proceedings of the VLDB (PVLDB), 2015.

[78] Radhika Mittal, Rachit Agrawal, Sylvia Ratnasamy, Scott Shenker. Universal Packet Scheduling. ACM Workshop on Hot Topics in Networks (HotNets), 2015.

[79] Rachit Agrawal, Anurag Khandelwal, Ion Stoica. Succinct: Enabling Queries on Compressed Data. Usenix Symposium on Networked Systems Design and Implementation (NSDI), 2015.

[80] Peter X Gao, Akshay Narayan, Gautam Kumar, Rachit Agarwal, Sylvia Ratnasamy, Scott Shenker. pHost: Distributed Near-Optimal Datacenter Transport over Commodity Network Fabric. CoNEXT 2015: International Conference on emerging Networking EXperiments and Technologies, 2015.

[81] Xinghao Pan, Dimitris Papailiopoulos, Sameet Oymak, Ben Recht, Kannan Ramchandran, Michael Jordan. Parallel Correlation Clustering on Big Graphs. Advances in Neural Information Processing Systems (NIPS), 2015.

[82] Kevin Jamieson, Lalit Jain, Chris Fernandez, Nick Glattard, Robert Nowak. NEXT: A System for Real-World Development, Evaluation, and Application of Active Learning. Neural Information Processing Systems, 2015.

[83] Anurag Khandelwal, Rachit Agarwal, Ion Stoica. BlowFish: Dynamic Storage-Performance Tradeoff in Data Stores. USENIX Symposium on Networked Systems Design and Implementation (NSDI), 2016.
https://www.usenix.org/conference/nsdi16/technical-sessions/presentation/khandelwal

[84] Xiangrui Meng, Joseph Bradley, Burak Yavuz, Evan Sparks, Shivaram Venkataraman, Davies Liu, Jeremy Freeman, DB Tsai, Manish Amde, Sean Owen, Doris Xin, Reynold Xin, Michael Franklin, Reza Zadeh, Matei Zaharia, Ameet Talwalkar. MLlib: Machine Learning in Apache Spark. Journal of Machine Learning Research, 2016.

[85] Philipp Moritz, Robert Nishihara, Michael Jordan. A Linearly-Convergent Stochastic L-BFGS Algorithm. Artificial Intelligence and Statistics (AISTATS), 2016.

[86] Philipp Moritz, Robert Nishihara, Ion Stoica, Michael Jordan. SparkNet: Training Deep Networks on Spark. International Conference on Learning Representations (ICLR), 2016.

[87] Adithya Murali, Animesh Garg, Sanjay Krishnan, Florian Pokorny, Pieter Abbeel, Ken Goldberg. TSC-DL: Unsupervised Trajectory Segmentation of Multi-Modal Surgical Demonstrations with Deep Learning. ICRA, 2016.

[88] Qifan Pu, Haoyuan Li, Matei Zaharia, Ali Ghodsi, Ion Stoica. FairRide: Near-Optimal, Fair Cache Sharing. USENIX Symposium on Networked Systems Design and Implementation (NSDI), 2016.

[89] Anand Padmanabha Iyer, Li Erran Li, Tathagata Das, Ion Stoica. Time-Evolving Graph Processing at Scale. Graph Data-management Experiences & Systems (GRADES), 2016.

[90] Xu Chu, Ihab Ilyas, Sanjay Krishnan, Jiannan Wang. Data Cleaning: Overview and Emerging Challenges. SIGMOD Tutorial, 2016.

[91] Frank Austin Nothaft, David Patterson, Anthony Joseph. Rapid and Efficient Analysis of 20,000 RNS-seq Samples with Toil. bioRxiv, 2016. http://biorxiv.org/content/early/2016/07/07/062497

[92] Zhao Zhang, Kyle Barbary, Frank Austin Nothaft, Evan Sparks, Oliver Zahn, Michael Franklin, David Patterson, Saul Perlmutter. Kira: Processing Astronomy Imagery Using Big Data Technology. IEEE Transaction on Big Data, 2016.

[93] Reza Bosagh Zadeh, Xiangrui Meng, Alexander Ulanov, Burak Yavuz, Li Pu, Shivaram Venkataraman, Evan Sparks, Aaron Staple, Matei Zaharia. Matrix Computations and Optimization in Apache Spark. 22nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2016. https://amplab.cs.berkeley.edu/publication/matrix-computations-and-optimization-in-apache-spark/

[94] Sanjay Krishnan, Jiannan Wang, Eugene Wu, Michael Franklin, Ken Goldberg. ActiveClean: Interactive Data Cleaning For Statistical Modeling. 42nd International Conference on Very Large Data Bases, 2016. http://www.vldb.org/pvldb/vol9/p948-krishnan.pdf

[95] Michael Laskey, Caleb Chuck, Jonathan Lee, Jeff Mahler, Sanjay Krishnan, Kevin Jamieson, Anca Dragan, Ken Goldberg. Comparing Human-Centric and Robot-Centric Sampling for Robot Deep Learning from Demonstrations. Arvix, 2016. https://arxiv.org/abs/1610.00850

[96] Xinghao Pan, Maximilian Lam, Stephen Tu, Dimitris Papailiopoulos, Ce Zhang, Michael Jordan, Kannan Ramchandran, Chris Re, Ben Recht. CYCLADES: Conflict-free Asynchronous Machine Learning. Advances in Neural Information Processing Systems (NIPS), 2016.

[97] Evan Sparks, Shivaram Venkataraman, Tomer Kaftan, Michael Franklin, Ben Recht. KeystoneML: Optimizing Pipelines for Large-Scale Advanced Analytics. ICDE 2017: IEEE International Conference on Data Engineering 2017, 2017.

LIST OF SYMBOLS, ABBREVIATIONS, AND ACRONYMS

BDAS	Berkeley Data Analysis System
AMPLab	Algorithms, Machines, and People Lab
WSCs	warehouse-scale computers
TB	Terabyte
PB	Petabyte
GB	Gigabyte
IDC	International Data Corporation
ML	Machine learning
G-OLA	Generalized Online Aggregation for Interactive Analysis
S3	Amazon Simple Storage Service
DAG	directed acyclic graph
RDD	Resilient Distributed Datasets
HDFS	Hadoop Distributed File System
URL	Uniform Resource Locator
API	Application programming interface
SQL	Structured Query Language
NoSQL	Non Structured Query Language
EDW	Enterprise Data Warehouse
ETL	Extract, Transform, Load
HiveQL	Hive Query Language
RDBMS	Relational Database Management System
UDFs	User-defined functions
OSS	Object Storage Service
EMC	EMC Corporation
SSD	Solid State Device
DRAM	Dynamic random-access memory
CoCoA	A Framework for Distributed Optimization
SNAP	Scalable Nucleotide Alignment Program
ADAM	Genomics Formats and Processing Patterns for Cloud Scale Computing
SMaSH	A Benchmarking Toolkit for Human Genome Variant Calling
CS	Computer Science
GPU	Graphics processing unit
BWA	Bowtie2 and SOAP2.
GraphX	Apache Spark's API for graphs and graph-parallel computation
SIGMOD	Special Interest Group on Management of Data
OSDI	Operating Systems: Design and Implementation
MDCC	Multi-Data Center Consistency
PIQL	a SQL like language
SAQP	Sampling-based approximate query processing
L-BFGS	Limited-memory Broyden–Fletcher–Goldfarb–Shanno algorithm
I/O	Input/Output

RAM	Random Access Memory
USENIX	The Advanced Computing Systems Association
HotDep	International Conference on Hot Topics in System Dependability
CIDR	International Conference on Innovative Data Systems Research
KDD	International Conference on
SOSP	International Conference on
ICDM	International Conference on
CHI	International Conference on Computer-Human Interaction
ACM	Association for Computing Machinery
LISA	Large Installation System Administration Conference
SOCC	Symposium on Cloud Computing
NIPS	Neural Information Processing Systems
NSDI	Symposium on Networked Systems Design and Implementation
SoCC	Symposium on Cloud Computing
ISRR	The International Symposium on Robotics Research
IEEE	Institute of Electrical and Electronics Engineers
VLDB	Very Large Data Bases
PBLDB	Proceedings of the VLDB
AISTATS	International Conference on Artificial Intelligence and Statistics
ICLR	International Conference on Learning Representations
ICRA	International Conference on Robotics and Automation
GRADES	Graph Data-management Experiences and Systems